CSci 4271W
Development of Secure Software Systems
Day 26: Authentication. ethics, and law

Stephen McCamant

University of Minnesota, Computer Science & Engineering

Outline

- ROC curve example
- Web authentication
- Names and identities
- Ethics and security

Extreme biometrics examples

- `exact_iris_code_match`: very low false positive (false authentication)
- `similar_voice_pitch`: very low false negative (false reject)

Where are these in ROC space?

A `if (iris()) return REJECT; else return ACCEPT;`

B `return REJECT;`

C `if (iris()) return ACCEPT; else return REJECT;`

D `if (iris() && pitch()) return ACCEPT; else return REJECT;`

E `return ACCEPT;`

F `if (rand() & 1) return ACCEPT; else return REJECT;`

G `if (pitch()) return ACCEPT; else return REJECT;`

H `if (iris() || pitch()) return ACCEPT; else return REJECT;`

Outline

- ROC curve example
- Web authentication
- Names and identities
- Ethics and security

Per-website authentication

- Many web sites implement their own login systems
  - \+ If users pick unique passwords, little systemic risk
  - – Inconvenient, many will reuse passwords
  - – Lots of functionality each site must implement correctly
  - – Without enough framework support, many possible pitfalls

Building a session

- HTTP was originally stateless, but many sites want stateful login sessions
- Built by tying requests together with a shared session ID
- Must protect confidentiality and integrity

Session ID: what

- Must not be predictable
  - Not a sequential counter
- Should ensure freshness
  - E.g., limited validity window
- If encoding data in ID, must be unforgeable
  - E.g., data with properly used MAC
  - Negative example: crypt(username $\|$ server secret)

Session ID: where

- Session IDs in URLs are prone to leaking
  - Including via user cut-and-paste
- Usual choice: non-persistent cookie
  - Against network attacker, must send only under HTTPS
- Because of CSRF, should also have a non-cookie unique ID

Session management

- Create new session ID on each login
- Invalidate session on logout
- Invalidate after timeout
  - Usability / security tradeoff
  - Needed to protect users who fail to log out from public browsers

Account management

- Limitations on account creation
  - CAPTCHA? Outside email address?
- See previous discussion on hashed password storage
- Automated password recovery
  - Usually a weak spot
  - But, practically required for large system

Client and server checks

- For usability, interface should show what's possible
- But must not rely on client to perform checks
- Attackers can read/modify anything on the client side
- Easy example: item price in hidden field

Direct object references

- Seems convenient: query parameter names resource directly
  - E.g., database key, filename (path traversal)
- Easy to forget to validate on each use
- Alternative: indirect reference like per-session table
  - Not fundamentally more secure, but harder to forget check

Function-level access control

- E.g. pages accessed by URLs or interface buttons
- Must check each time that user is authorized
  - Attack: find URL when authorized, reuse when logged off
- Helped by consistent structure in code

Outline

ROC curve example

Web authentication

Names and identities

Ethics and security

Accounts versus identities

- "Identity" is a broad term that can refer to a personal conception or an automated sytem
- "Name" is also ambiguous in this way
- "Account" and "authentication" refer unambiguously to institutional/computer abstractions
- Any account system is only an approximation of the real world

Real human names are messy

- Most assumptions your code might make will fail for someone
  - ASCII, length limit, uniqueness, unchanging, etc.
- So, don't design in assumptions about real names
- Use something more computer-friendly as the core identifier
  - Make "real" names or nicknames a presentation aspect

Zooko's triangle

- Claims (2001) it is hard/impossible for a naming scheme to be simultaneously:
  - Human-meaningful
  - Secure
  - Decentralized
- Too imprecise to be definitively proven/refuted
  - Blockchain-based name systems are highest-profile claimed counterexamples
- A useful heuristic for seeing design tensions

Identity documents: mostly unhelpful

- "Send us a scan of your driver's license"
  - Sometimes called for by specific regulations
  - Unnecessary storage is a disclosure risk
  - Fake IDs are very common

Identity numbers: mostly unhelpful

- Common US example: social security number
- Variously used as an identifier or an authenticator
  - Dual use is itself a cause for concern
- Known by many third parties (e.g., banks)
- No checksum, guessing risks
- Published soon after a person dies

"Identity theft"

- The first-order crime is impersonation fraud between two other parties
  - E.g., criminal trying to get money from a bank under false pretenses
- The impersonated "victim" is effectively victimized by follow-on false statements
  - E.g., by credit reporting agencies
  - These costs are arguably the result of poor regulatory choices
- Be careful w/ negative info from 3rd parties

Outline

ROC curve example

Web authentication

Names and identities

Ethics and security

Don't be evil

- Broadly, ethics are principles for distinguishing good from bad actions
- Most people try to be good most of the time
  - But there are hard cases
- Topics important enough for security are usually also important for ethics
  - But adversaries often arise from ethical disagreement

Principles and consequences

- Ethical reasoning tends to be a mix of:
- Principles for categorizing actions as good or bad
  - Religions and laws provide many examples
- Attention to the consequences of actions
  - E.g., actions are evil because of their negative effects
- Another meta-principle: people's ethical intuitions vary

Ethics and laws

- The legal system is a primary way societies enforce ethical guidelines
  - But the law is an imperfect consensus approximation of ethics
- Following the law and being ethical can be separate constraints
  - You should try to satisfy both

Beyond white and black hats

- In describing techniques, we posit a clear distinction of attackers and defenders
- But in real scenarios, you can't assume that attacker = bad and defender = good
- What follows are some specific situations showing more complexity

Ethics of security research

- Why do good people research (and teach) about attack techniques?
  1. In order to effectively defend, you have to be able to anticipate attacker strategies
  2. In some cases, attacks may be ethically justified
- Common example: finding vulnerabilities so they can be fixed

Responsible disclosure

- If you find a vulnerability in software, who should you tell about it? Two extremes:
  - Only the author/vendor ever needs to know
  - Make the information fully public right away (full disclosure)
- Security researchers often push on vendors for more and faster disclosure
- A common compromise is to give vendors a head start, but with a deadline
  - E.g., Google uses 90 days (or 7 days if being used)

Nation states

- Many governments would argue they need to break the security of criminals or foreign spies
  - "justice", "public safety", "national security", etc.
- "Cyber-warfare" has both offensive and defensive aspects
  - Compare with various ethical perspectives on killing in war

Interoperability and repair

- Vendors of devices can have economic desires to control how the devices interact with other devices or can be repaired
  - Classic example: expensive proprietary ink cartridges
- If vendors use security and cryptography techniques to implement these restrictions, is it ethical to attack them?

Copy protection and DRM

- Vendors of software and media would prefer you can't make copies to give to your friends
  - Many generations of attempts to implement such restrictions
  - Fundamentally hard, because the data must be decoded to be used
  - Keeping software from being reverse engineered is also hard
- Do the ethics depend on how competent the technique is?

Malware analysis

- Labeling software as malicious is defining it to be the evil side
  - E.g., viruses, botnet clients
- Leads to many software security concerns being inverted
- Preventing reverse engineering is a common goal of DRM software and malware